Amendments to the Specification:

Please replace paragraph **[0003]** with the following amended paragraph:

[0003] JAVASERVER™ Pages is the JAVA™ programming language platform technology (hereinafter JAVA™ platform) for building applications containing dynamic Web content such as HTML, DHTML, XHTML, and XML. A JAVASERVER™ Page (JSP) is a text-based document that describes how to process a request to create a response. The description inter-mixes template data with some dynamic actions, taking advantage of the capabilities of the JAVA™ platform. The template data is commonly fragments of a structured document (HTML, DHTML, XHTML or XML), and the dynamic actions can be described with scripting elements and/or server-side action tags.

Please replace paragraph **[0004]** with the following amended paragraph:

[0004] A simple example of a JSP page is shown in Figure 2. The example shows the response page, which is intended to be a short list of the day of the month and year, at the moment the request is received by the server. The page itself contains some fixed template text, and JSP elements that are shown underlined in the figure. The underlined actions are executed on the server side. When a client makes a request, such as an HTTP request, a request object requests a response from the JAVASERVER™ container object. The first element creates a JAVA™ Bean component named clock, of type calendar.jspCalendar. The next two elements use the JAVA™ Bean component to display some of its properties (i.e. month and year). The output is sent to a response object which sends a response back to the client.

Please replace paragraph **[0007]** with the following amended paragraph:

[0007] In standard implementations, a JSP page is translated into JAVA™ programming language code (hereinafter JAVA™ code) that runs on a server. The transformation from the JSP page into JAVA™ code is done once, when the page is first accessed. The JAVA™ code running on the server is activated when a request is received. In order to create a response object, certain data is passed verbatim to the response, and some actions are executed on the server side. Some of these actions are explicitly spelled out in the JSP page, and other actions are described by the semantics of the text. Thus, there are three types of code: verbatim code,

scripting code from the JSP page, and code which has to be defined by the tag library. As used herein, "tags" in the JSP context will be referred to as "actions."

Please replace paragraph **[0013]** with the following amended paragraph:

[0013] Attached to each action, there is a TagExtraInfo class that describes the action. The TagExtraInfo method will return at translation time an array of objects that describe the run-time effect of the action. Each object contains a name of each variable as a valid JAVA™ programming language object (hereinafter JAVA™ object), a JAVA™ programming language type (hereinafter JAVA™ type) of each variable, and a scope parameter which defines a variable's scope with the page and indicates whether the variable is new or pre-existing. At translation-time, the JSP page is translated into a servlet class. In the process of translation, the translator asks for the list of scripting variables from TagExtraInfo object for a given action. If a new variable has been defined, the translator may need to provide a new declaration statement. At run time, the action tag handler references a pageContext object, which provides key values for specified variable names. The pageContext object points to an object associated with a given variable name and is created when the page is executed, thereby associating a variable with a value.

Please replace paragraph **[0028]** with the following amended paragraph:

[0028] As the term is used herein, a tag extension mechanism is a specialized sub-language that enables the addition of new or custom actions, thus allowing the JSP "language" to be easily extended in a portable fashion. A typical example would be elements to support embedded database queries. Tag libraries can be used by JSP authors or JSP authoring tools and can be distributed along with JSP pages to any JSP container (i.e. JSP environment and/or engine), such as Web and application servers. The tag extension mechanism of the present invention can be used from JSP pages written using any valid scripting language, although the mechanism itself only assumes a JAVA™ programming language RunTime environment.

Please replace paragraph **[0033]** with the following amended paragraph:

[0033] As described above, the JSP container (translator) knows the details of the scripting language of the page, whereas the author of the action tag is unaware of which scripting language will be used. So in order to provide for maximum flexibility and portability, the

present invention provides a mechanism that insulates the tag implementation (tag mechanism) from the scripting language details. If both the tag library and the scripting page were written in the same language, such as the JAVA™ programming language, the variables could be synchronized more easily. However, in order to provide a more general solution, the present invention exposes the variables in an action tag at translation time via the TagExtraInfo class, and uses a pageContext object to map the variables to values at run time.